

Amendments to the Claims: This listing of claims will replace all prior versions, and listings, of claims in the application

Listing of Claims:

1. (Original) A memory controller for managing memory requests from a plurality of requesters to a plurality of memory banks, the memory controller comprising:

an arbiter configured to receive the memory requests from the plurality of requesters, the arbiter assigning a first memory request to a first processing path and a second memory request to a second processing path responsive to the memory banks requested by the received and assigned memory requests;

a first path controller coupled to the arbiter and the plurality of memory banks, the first path controller configured to process the first memory request in the first processing path to activate a first memory bank associated with the first memory request;

a second path controller coupled to the arbiter and the plurality of memory banks, the second path controller configured to process the second memory request in the second processing path to activate a second memory bank associated with the second memory request while the first memory bank is active; and

a synchronizer coupled between the first path controller and the second path controller for synchronizing the first and second path controllers such that the first and second memory requests processed by the first and second path controllers, respectively, do not conflict.

2. (Original) The memory controller of claim 1, wherein the arbiter, the first path controller, the second path controller, and the synchronizer are implemented as a single field programmable gate array.

3. (Original) The memory controller of claim 1, wherein the arbiter, the first path controller, the second path controller, and the synchronizer are configured for use with an SDRAM memory device comprising the first and second memory banks.

4. (Original) The memory controller of claim 1, wherein the first path controller comprises at least:

first path circuitry that passes addresses and data associated with the first memory request;

a first path timing controller that controls the first path circuitry and activates the first memory bank associated with the first memory request; and

address and data multiplexers that multiplex addresses and data associated with the first memory request for interfacing with the memory banks; and

wherein the second path controller comprises at least:

second path circuitry that passes addresses and data associated with the second memory request;

a second path timing controller that controls the second path circuitry and activates the second memory bank associated with the second memory request; and

the address and data multiplexers that multiplex addresses and data associated with the first memory request, the address and data multiplexers further multiplexing addresses and data associated with the second memory request for interfacing with the memory banks.

5. (Original) The memory controller of claim 1, wherein the synchronizer comprises:

delay circuits coupled between the first and second path controllers to set delay values therebetween to adjust the timing of the first and second path controllers during processing of the first and second memory requests responsive to the first and second memory requests.

6. (Original) The memory controller of claim 1, wherein the first path controller is further configured to initialize and refresh the plurality of memory banks.

7. (Currently Amended) The memory controller of claim 1, wherein the arbiter assigns the first and second memory requests using a fairness algorithm, the fairness algorithm comparing the plurality of requesters to a grant history register, identifying ~~ones off~~from the plurality of requesters those requesters that have had previous memory requests granted during a current

arbitration cycle, and identifying the first memory request by a first memory requester from the plurality of memory requesters not on the grant history register and not having a current request by the second path controller using fixed priority logic.

8. (Currently Amended) The memory controller of claim 1, wherein the arbiter assigns the second memory request to the second path controller when the first path controller is active if the first and second memory banks are not equal the same memory bank.

9. (Original) A method for managing memory requests received from a plurality of requesters to access a plurality of memory banks, the method comprising the steps of:

assigning a first memory request to a first processing path and a second memory request to a second processing path responsive to the memory banks requested by the received and assigned memory requests;

processing the first memory request in the first processing path and the second memory request in the second processing path such that the first memory request activates a first memory bank and the second memory request activates a second memory bank while the first memory bank is active; and

synchronizing the processing in the first and second processing paths such that the first and second memory requests processed in the first and second paths, respectively, do not conflict.

10. (Original) The method of claim 9, further comprising the step of:

multiplexing data and addresses associated with the first and second memory requests for accessing the plurality of memory banks.

11. (Original) The method of claim 9, wherein the processing step comprises generating a read command or a write command in each of the first and second processing paths and wherein the synchronizing step comprises:

delaying processing of the second memory request with respect to the first memory request responsive to the read/write commands of the first and second memory requests to concatenate the first and second memory requests.

12. (Currently Amended) The method of claim 9, further comprising the step of:

~~initializing~~ initializing the plurality of memory banks using the first processing path.

13. (Currently Amended) The method of claim 9, wherein the assigning step comprises:

receiving the memory requests from the plurality of memory requesters during a current arbitration cycle;

comparing the plurality of memory requesters to a grant history register identifying ~~ones of from~~ the plurality of memory requesters those requesters that have had previous memory requests granted during the current arbitration cycle;

identifying the first memory request by a first memory requester from the plurality of memory requesters not on the grant history register and not having a current request in the second processing path using fixed priority logic; and

adding the first memory requester to the grant history register.

14. (Currently Amended) The method of claim 9, wherein the second memory request is assigned to the second processing path when the first processing path is in use if the first and second memory banks are not equal the same memory bank.

15. (Original) A system for managing memory requests from a plurality of requesters to a plurality of memory banks comprising:

means for assigning a first memory request to a first processing path and a second memory request to a second processing path responsive to the memory banks requested by the received and assigned memory requests;

means for processing the first memory request in the first processing path and the second memory request in the second processing path such that the first memory request activates a first memory bank and the second memory request activates a second memory bank while the first memory bank is active; and

means for synchronizing the processing in the first and second processing paths such that the first and second memory requests processed in the first and second paths, respectively, do not conflict.

16. (Original) The system of claim 15, further comprising:

means for combining the first and second memory requests for accessing the plurality of memory banks, wherein the processing means comprises generating a read command or a write command in each of the first and second processing paths and wherein the commands are concatenated by the synchronizing and combining means.

17. (Original) An arbitration method for assigning at least one controller to manage a plurality of memory requests from a plurality of requesters to a memory device having at least one memory bank, each memory request associated with a unique requester, the method comprising the steps of:

receiving the plurality of memory requests from the plurality of memory requesters during a current arbitration cycle;

comparing the plurality of memory requesters to a grant history register identifying ones of the plurality of memory requesters that have had previous memory requests granted during the current arbitration cycle;

assigning a memory request to one of the at least one controllers from the plurality of memory requesters not in the grant history register using fixed priority logic; and

adding the requester of the assigned memory request to the grant history register.

18. (Original) The method of claim 17, wherein each requester is assigned a unique value and wherein the fixed priority logic performs the step of:

identifying from the plurality of memory requesters not on the grant history register a lowest memory requester having a lowest value among the plurality of memory requesters not on the grant history register for assignment to one of the at least one controller.

19. (Original) The method of claim 17, wherein the at least one controller comprises a first path timing controller and a second path timing controller and wherein the method further comprises the steps of:

receiving a first processing indicator at the first path timing controller representing a first processing requester of a first processing memory request being processed at the second path timing controller; and

receiving a second processing indicator at the second path timing controller representing a second processing requester of a second processing memory request being processed at the first path timing controller;

wherein the step of assigning the memory request comprises assigning a first memory request to the first path timing controller from the plurality of memory requesters not on the grant history register and not being processed by the second path controller using fixed priority logic and assigning a second memory request to the second path timing controller from the plurality of memory requesters not on the grant history register and not being processed by the first path timing controller using fixed priority logic.

20. (Original) The method of claim 19, wherein each requester is assigned a unique value and wherein the fixed priority logic comprises the steps of:

identifying from the plurality of memory requesters not on the grant history register and not being processed by the second path timing controller a first lowest memory requester having a first lowest value among the plurality of memory requesters not on the grant history register and not being processed by the second path timing controller that is requesting access to a first memory bank not currently activated by the second path timing controller.

identifying from the plurality of memory requesters not on the grant history register and not being processed by the first path timing controller a second lowest memory requester having a second lowest value among the plurality of memory requesters not on the grant history register and not being processed by the first path timing controller that is requesting access to a second memory bank not currently activated by the first path timing controller.

21. (Original) A memory controller for managing memory requests from a plurality of requesters to a plurality of memory banks over data, address, and command busses comprising:

an arbiter configured to receive the plurality of memory requests from the plurality of memory requesters during a current arbitration cycle, compare the plurality of memory requesters to a grant history register to identify ones of the plurality of memory requesters that have had previous memory requests granted during the current arbitration cycle, and to identify a memory request from the plurality of memory requesters not on the grant history register using fixed priority logic; and

at least one path controller coupled to the arbiter and the plurality of memory banks, the at least one path controller configured to process the identified memory request.

22. (Original) The memory controller of claims 21, wherein the arbiter and the at least one path controller are implemented in a field programmable gate array.